Vorwort

Seit einigen Jahren gibt es Qualitätsmanagement- und Qualitätssicherungsnormen für die Entwicklung, Lieferung und Wartung von Software (z. B. DIN ISO 9000–3) und ein fünfstufiges Reifegradmodell zur Bewertung des Entwicklungsprozesses einer Institution. Viele Software-Firmen und -Abteilungen haben sich seitdem ihren normgerechten Entwicklungsstandard zertifizieren lassen. Dennoch hat es keinen bemerkenswerten Sprung in der Qualität und Fehlerfreiheit der Softwareprodukte gegeben. Dies liegt vor allem daran, daß die oben zitierte DIN nur die Qualität des Entwicklungsprozesses und nicht die Qualität der erstellten *Produkte* im Auge hat und daß die meisten Institutionen noch nicht die beiden obersten Stufen des Reifegradmodells erreicht haben, bei denen erstmals quantitative und qualitative Produkteigenschaften betrachtet werden.
Daher konzentriert sich dieses Buch auf die Überprüfung entsprechender Produkteigenschaften, d. h. es werden Methoden und Verfahren vorgestellt, die Abweichungen (Fehler) aufdecken und beseitigen können. Die grundlegenden Konzepte werden durch entsprechende Problemstellungen motiviert, meistens formal definiert und umgangssprachlich und anhand von Beispielen und Abbildungen erläutert. Außerdem werden Angaben zu Vor- und Nachteilen (Kosten und Nutzen) der Testmethoden gemacht, mögliche Algorithmen werden vorgestellt und Eigenschaften der Verfahren werden bewiesen, wenn das im gegebenen Rahmen möglich ist, ansonsten wird auf entsprechende Quellen verwiesen.

Leserinnen und Leser — seien es Studierende an Universitäten oder Fachhochschulen, Personen in der Forschung oder in der Praxis (Software-Entwicklung oder -Management) — können dieses Buch daher zum Selbststudium verwenden und die dargestellten Methoden als Grundlage für die (Weiter-)Entwicklung entsprechender Werkzeuge verwenden. (Auf existierende Werkzeuge und Prototypen wird in diesem Buch nicht eingegangen, da dies den Rahmen sprengen würde und Werkzeuge schneller veralten als Methoden.)

Das Buch gliedert sich in vier Teile. Der einführende Teil I beginnt mit zwei Beispielen für den Programmtest. Es schließen sich zwei Kapitel an, die einen Überblick über die grundlegenden Problemstellungen und Lösungsansätze beim Testen geben. Teil II stellt in Kapitel 4 die „klassischen“ unsystematischen, datenbereichsbezogenen und funktionsbezogenen Testmethoden vor. Neuere Konzepte zum Testen auf der Basis von Pfadausdrücken und algebraischen Spezifikationen enthält Kapitel 5. Kapitel 6 widmet sich dem Vergleich und der Bewertung dieser Testmethoden.
Die Methoden in Teil II kommen überwiegend beim spezifikationsorientierten Testen

vor. Sie können aber teilweise (in abgewandelter Form) auch beim implementationsorientierten Testen verwendet werden, das in Teil III abgehandelt wird. Dazu gehören Testmethoden, die sich am Kontrollfluß (Kapitel 7), am Datenfluß (Kapitel 8) oder an den Ausdrücken, Anweisungen und Daten (Kapitel 9) orientieren. Die Kapitel 10 und 11 vergleichen Kosten und Nutzen dieser Testmethoden.
Weitere Aspekte des Testens werden in Teil IV behandelt. In Kapitel 12 werden die — früh einsetzbaren — Techniken der statischen Analyse und die bisher nur theoretisch interessanten Techniken der symbolischen Ausführung und Verifikation vorgestellt. Kapitel 13 behandelt das praktisch relevante Testen von großen Programmsystemen durch Integrations- und Systemtests. Kapitel 14 stellt die neuartigen Probleme und Lösungsansätze zum Testen nebenläufiger Systeme vor. Da das Testen nur eine Diagnose liefert, die Therapie in Form der Fehlerbehebung aber das eigentliche Ziel ist, werden entsprechende Techniken in Kapitel 15 vorgestellt. Das Management des Testens auf der Grundlage von Komplexitätsmaßen für die (Test-)Aufwandsermittlung und die Auswahl von Testmethoden, die kombinierte Verwendung von Testmethoden sowie die Abschätzung der Restfehler und der Zuverlässigkeit sind Inhalte von Kapitel 16. Kapitel 17 faßt die Ergebnisse des Buches kurz zusammen und gibt einen Ausblick auf neue Herausforderungen für Softwaretestmethoden. Für weitergehende Studien ist eine Aufstellung von einschlägigen Zeitschriften und Standards sowie ein umfangreiches Literaturverzeichnis im Anhang zu finden.

Dozenten können jeden der vier Teile dieses Buches in ca. 24 Stunden präsentieren (z. B. im Sommersemester mit 12 Wochen à zwei Stunden pro Woche), wobei Teil I zusammen mit Kapitel 12 vorgetragen werden sollte.

Dieses Buch wurde aus mehreren Quellen gespeist. Aufgrund von Anregungen durch meine einjährige Tätigkeit in einem Dortmunder Softwarehaus habe ich im Wintersemester 1984/85 das Thema Softwaretesten zum Inhalt einer Vorlesung an der Universität Dortmund gemacht, zu der mein Kollege Herbert Schippers die Übungen beisteuerte. In den folgenden Jahren haben wir weitere Erfahrungen mit dem Einsatz und der Entwicklung von Testmethoden und Testwerkzeugen in Projekten, studentischen Projektgruppen und Diplomarbeiten gemacht. Auf Anregung des Teubner-Verlags entwickelten Herbert Schippers und ich Ende 1989 ein Konzept für eine erste Version eines Buches auf Grundlage des Vorlesungsskripts „Softwaretestmethoden“. Herbert Schippers hat die erste Fassung der Kapitel 3 und 4 geschrieben, die anderen Kapitel wurden von mir beigesteuert und als Grundlage für entsprechende Vorlesungen vom Wintersemester 1990/91 bis zum Sommersemester 1996 verwendet und parallel dazu von mir überarbeitet und ergänzt. Für die vorliegende Fassung des Textes trage ich — nach dem Ausscheiden von Herbert Schippers aus dem „Buchprojekt“ Ende 1990 — die alleinige Verantwortung.

Zu Stil und Rechtschreibung dieses Buches sind fünf Anmerkungen angebracht:

1. Die verwendeten Begriffe werden meistens in (numerierten) Definitionen explizit eingeführt oder aber durch Fettdruck hervorgehoben. Diese „Definitionen“ sind nicht immer wie formale Definitionen im mathematischen Sinne zu verstehen, sondern oft als informelle, möglichst konkrete Begriffsbeschreibungen zu sehen. Ebenso verhält es sich mit Aussagen über Testmethoden, die teilweise als (numerierte) „Sätze“ formuliert sind.

2. Da dieses Buch in deutscher Sprache geschrieben ist, werden — wenn immer möglich — deutsche Begriffe verwendet. Bei unüblichen deutschen Begriffen wird die englische Bezeichnung als Erläuterung in Klammern hinzugefügt.

3. Zur Behandlung des Problems mit den „männlichen“ Berufsbezeichnungen gibt es bisher mehrere Ansätze:

   (a) einmaliger Hinweis darauf, daß z. B. mit „Softwaretester“ auch Frauen gemeint sind,

   (b) konsequente Verwendung der männlichen und weiblichen Form (z. B. Softwaretester und Softwaretesterin),

   (c) neue Form mit großem „I“ (z. B. SoftwaretesterIn),

   (d) Verwendung der rein weiblichen Form.

   Da diese Ansätze alle nicht befriedigen, habe ich mich für einen neuen Ansatz entschieden: In den Kapiteln 2 und 12 wird die rein weibliche oder eine neutrale Form, in den anderen Kapiteln nur die männliche oder eine neutrale Form verwendet.

4. Hervorhebungen werden folgendermaßen verwendet: **Fettdruck** für definierte Begriffe, *Kursivdruck* für die Betonung einzelner Wörter (als Lesehilfe) bzw. durchgängig zur Abgrenzung von Beispielen, Unterstreichungen als Gliederungshilfe, um Unterpunkte abzugrenzen, und in durchgängig kursiv geschriebenen Texten als Hervorhebung einzelner Wörter.

5. Die Rechtschreibung orientiert sich noch nicht an den neuen Rechtschreibregeln von 1997, daher wird z. B. das Wort „Testen“ in der Form „Te–sten“ und nicht als „Tes–ten“ getrennt.

Dieses Buch wäre nicht zustande gekommen ohne die wohlwollende Unterstützung und Duldung durch meinen unmittelbaren Vorgesetzten, Prof. Dr. Bernd Reusch, und meine Familienmitglieder Bärbel, Sonja und Inga, die mir erlaubt haben, einen Teil meiner Arbeits- und Freizeit für dieses Buch zu verwenden. Zu danken habe ich auch dem Herausgeber, Prof. Dr. Volker Claus, und dem Lektor im Teubner-Verlag, Dr. Peter Spuhler, für ihre Anregungen, ihre Geduld und insbesondere für

ihr Verständnis für die Verzögerungen bei der Erstellung des Buches.
Vielen meiner (Ex-)Kolleginnen und Kollegen habe ich für ihre kritischen und konstruktiven Bemerkungen und Hinweise auf Fehler in einzelnen Kapiteln von Vorversionen des Buchtextes zu danken, insbesondere Wolfgang Ditt, Beate Fricke, Arnulf Mester und meinem GI-Kollegen Andreas Spillner. Ebenso danke ich meinen Diplomanden und den Hörerinnen und Hörern meiner Vorlesungen, André Deparade, Nicola Engel und Martin Pfeiffer, insbesondere aber Norbert Feldhaus und Uwe Tegethoff. Für die letzte Version des vollständigen Buchtextes haben sich dankenswerterweise Uli Mette und Andreas Wieck als Korrekturleser angeboten.

Eine Vorversion des Textes hat Uwe Halfter angefertigt (in Wordplus auf einem Atari). Die endgültige Version des Textes und die Bilder wurden von Thomas Biskup in den letzten drei Jahren in unermüdlichem, nicht mit Geld zu bezahlendem Einsatz erstellt, außerdem hat er mit einer Fülle von kritischen Anmerkungen und Vorschlägen zur erheblichen Verbesserung der Qualität des Buches beigetragen.

Die Texte für dieses Buch wurden unter den Betriebssystemen Linux bzw. SunOS mit den Editoren „GNU Emacs“ und Jove in Form von 21 Dateien erstellt, außerdem wurden mit Xfig 93 Dateien mit Abbildungen und Tabellen erzeugt. Der Platzbedarf dieser 114 Dateien beträgt ca. 2,2 MByte. Für das Sachverzeichnis wurde von Thomas Biskup ein eigenes Programm namens `MakeIndex` (in perl) geschrieben, da das existierende Programm (für die Behandlung deutscher Umlaute) regelmäßig mit einem Fehler (segmentation fault) vorzeitig beendet wurde.
Mit Hilfe von zwei Dateien mit allgemeinen Formatierungsdefinitionen (teub_def.tex und zusatz.tex) wurde daraus mit dem Übersetzer für LaTeX2e eine Datei im dvi-Format erzeugt, die anschließend in das postscript-Format umgewandelt und auf einem Laser-Drucker ausgedruckt wurde.

Das „Testproblem“ für einen Text ist mindestens so schwierig wie für ein Programm, da die benutzten Textverarbeitungswerkzeuge mehr Fehler durchgehen lassen als die Übersetzer (Compiler) von Programmiersprachen. Einige Tippfehler oder auch gedankliche Fehler werden das Korrekturlesen (die „Text-Inspektion“, vgl. Kapitel 12) vermutlich ebenfalls überlebt haben, obwohl viele „Inspekteure“ (s. oben) beteiligt waren. Ich bin daher — wie die potentiellen Leser der nächsten Auflage dieses Buches — dankbar für jeden Korrekturvorschlag an meine Dienstadresse:

Universität Dortmund, Informatik I, Postfach 500 500, 44221 Dortmund
E-Mail: `riedeman@@ls1.informatik.uni-dortmund.de`

Dortmund, im Juli 1997 — Eike Hagen Riedemann

Inhalt